AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A network device driver architecture for enabling access between operating system kernel space and a network interface controller (NIC) as well as between user space and said NIC, comprising:

a kernel-space device driver adapted for enabling access between kernel space and user space via a kernel-space-user-space interface; and

user-space device driver functionality adapted for enabling direct access between user space and said NIC via a user-space-NIC interface, and said user-space device driver functionality adapted for interconnecting said kernel-space-user-space interface and said user-space-NIC interface to enable integrated kernel-space access and user-space access to said NIC.

2. (Previously Presented) The network device driver architecture according to claim 1, wherein said kernel-space device driver is adapted for establishing said kernel-space-user-space interface in relation to said user-space device driver functionality.

3. (Previously Presented) The network device driver architecture according to claim 1, wherein said user-space device driver functionality is adapted for fetching pointer information, pointing to data in a common memory, from a memory buffer associated with one of said kernel-space-user-space interface and said user-space-NIC interface and inserting said pointer information into a memory buffer associated with the other of said interfaces, thereby interconnecting said kernel-space-user-space interface and said user-space-NIC interface.

4. (Previously Presented) The network device driver architecture according to claim 1, wherein each of said kernel-space-user-space interface and said user-space-NIC interface is associated with two memory buffers, a transmit buffer and a receive buffer.

5. (Previously Presented) The network device driver architecture according to claim 4, wherein, for outbound kernel-level protocol communication, said kernel-space device driver is adapted for inserting pointer information, pointing to data in a common memory, into the transmit buffer associated with said kernel-space-user-space interface, and said user-space device driver functionality is adapted for fetching said pointer information therefrom and inserting it into the transmit buffer associated with said user-space-NIC interface, and said NIC is adapted for fetching said pointer information from the transmit buffer associated with said user-space-NIC interface and for reading corresponding data from said common memory based on the obtained pointer information.

6. (Previously Presented) The network device driver architecture according to claim 4, wherein, for inbound kernel-level protocol communication, said NIC is adapted for inserting pointer information, pointing to data in a common memory, into the receive buffer associated with said user-space-NIC interface, and said user-space device driver functionality is adapted for fetching said pointer information from the receive buffer associated with said user-space-NIC interface and inserting it into the receive buffer associated with said kernel-space-user-space interface, and said kernel-space device driver is adapted for fetching said pointer information for transfer to a kernel-level protocol, which reads the corresponding data from said common memory based on the pointer information.

7. (Previously Presented) The network device driver architecture according to claim 1, wherein said user-space device driver functionality is configured for execution in application context of a user application.

8. (Previously Presented) The network device driver architecture according to claim 7, wherein said user-space device driver functionality is implemented as user-space library functionality.

9. (Previously Presented) The network device driver architecture according to claim 1, wherein said kernel-space device driver is operable for directly accessing said NIC via a kernel-space-NIC interface in a first operational mode, and operable for accessing said NIC via said kernel-space-user-space interface, said user-space device driver functionality and said user-space-NIC interface in a second operational mode.

10. (Previously Presented) The network device driver architecture according to claim 9, wherein said user-space device driver functionality is configured for execution in application context of a user application, and said kernel-space device driver is adapted to switch to said first operational mode in response to a user application failure.

11. (Previously Presented) The network device driver architecture according to claim 9 or 10, wherein said kernel-space device driver includes watchdog functionality for switching to said first operational mode if there is no call from said user-space device driver functionality for a predetermined period of time.

12. (Previously Presented) The network device driver architecture according to claim 9, wherein said kernel-space device driver comprises:

a kernel-space agent for managing said kernel-space-user-space interface;

a network device driver core operable for directly accessing said NIC in said first operational mode, and operable for routing outgoing data to said kernel space agent and for receiving incoming data from said kernel space agent in said second operational mode.

13. (Previously Presented) The network device driver architecture according to claim 12, wherein said user-space device driver functionality is configured for execution in application context of a user application, and said kernel-space agent is adapted to

respond to a user application failure by ordering said network device driver core to switch to said first operational mode.

14. (Currently Amended) The network device driver architecture according to claim 12 ~~or 13~~, wherein said kernel-space agent includes watchdog functionality for ordering said network device driver core to switch to said first operational mode if there is no call from the user-space device driver functionality for a predetermined period of time.

15. (Previously Presented) A system for enabling operating system kernel space access as well as user space access to a network interface controller (NIC), said system comprising means for integrated kernel-space access and user-space access over the same NIC.

16. (Previously Presented) The system according to claim 15, wherein said means for integrated kernel-space access and user-space access over the same NIC comprises:

means for direct access between user space and NIC; and

means for user-space tunneled access between kernel-space and said NIC.

17. (Previously Presented) A method for enabling access between operating system kernel space and a network interface controller (NIC) as well as between user space and said NIC, said method comprising the steps of:

enabling access between kernel space and user space via a kernel-space-user-space interface;

enabling direct access between user space and said NIC via a user-space-NIC interface; and

interconnecting said kernel-space-user-space interface and said user-space-NIC interface to enable user-space tunneled access between kernel-space and said NIC.

18. (Previously Presented) The method according to claim 17, wherein said interconnecting step comprises the steps of:

fetching pointer information, pointing to data in a common memory, from a memory buffer associated with one of said kernel-space-user-space interface and said user-space-NIC interface; and

inserting said pointer information into a memory buffer associated with another ~~the other~~ of said interfaces.

19. (Previously Presented) The method according to claim 17, wherein said NIC access functionality is distributed between a kernel-space device driver and user-space device driver functionality.

20. (Previously Presented) The method according to claim 19, further comprising, for outbound kernel-level protocol communication, the steps of said kernel-space device driver inserting pointer information, pointing to data in a common memory, into the transmit buffer associated with said kernel-space-user-space interface, and said user-space device driver functionality fetching said pointer information therefrom and inserting it into the transmit buffer associated with said user-space-NIC interface, and said NIC fetching said pointer information from the transmit buffer associated with said user-space-NIC interface and reading corresponding data from said common memory based on the obtained pointer information.

21. (Currently Amended) The method according to claim 19, further comprising, for inbound kernel-level protocol communication, the steps of:

said NIC inserting pointer information, pointing to data in a common memory, into a receive buffer associated with said user-space-NIC interface; ~~, and~~

said user-space device driver functionality fetching said pointer information from the receive buffer associated with said user-space-NIC interface and inserting it into the receive buffer associated with said kernel-space-user-space interface; ~~and~~

said kernel-space device driver fetching said pointer information for transfer to a kernel-level protocol, which reads the corresponding data from said common memory based on the pointer information.

22. (Previously Presented) The method according to claim 17, wherein said step of enabling direct access between user space and said NIC and said interconnecting step are executed in application context of a user application.

23. (Previously Presented) The method according to claim 22, wherein said step of enabling direct access between user space and said NIC and said interconnecting step are performed by user-space device driver functionality implemented as user-space library functionality.

24. (Currently Amended) The method according to claim 17, further comprising the steps of:

in a first operational mode of a kernel-space device driver, directly accessing said NIC from said kernel-space device driver via a kernel-space-NIC interface; and

in a second operational mode of said kernel-space device driver, accessing said NIC via the interconnected kernel-space-user-space interface and user-space-NIC interface.

25. (Previously Presented) The method according to claim 24, wherein said step of enabling direct access between user space and said NIC and said interconnecting step are executed in application context of a user application, and the operating system orders said kernel-space device driver to switch to said first operational mode in response to a user application failure.

26. (Currently Amended) The method according to claim 24 ~~or 25~~, further comprising the step of switching to said first operational mode if there is no user-space call to said kernel-space device driver for a predetermined period of time.

27. (Previously Presented) A method for enabling operating system kernel space access as well as user space access to a network interface controller (NIC), said method comprising the step of providing integrated kernel-space access and user-space access over the same NIC.

28. (Previously Presented) The method according to claim 27, wherein said step of providing integrated kernel-space access and user-space access over the same NIC comprises the steps of:

enabling direct access between user space and NIC; and

enabling user-space tunneled access between kernel-space and said NIC.

29. (Previously Presented) The method according to claim 27, wherein integrated kernel-space access and user-space access is provided over the same NIC port.